

## CLAIM AMENDMENTS

### Claim Amendment Summary

#### **Claims pending**

- Before this Amendment: Claims 1-42 and 45-46.
- After this Amendment: Claims 1-42 and 45-46.

**Non-Elected, Canceled, or Withdrawn claims:** none.

**Amended claims:** none.

**New claims:** none.

---

### Claims:

1. (ORIGINAL) A kernel emulator for non-native program modules, the emulator comprising:

an interceptor configured to intercept kernel calls from non-native program modules;

a call-converter configured to convert non-native kernel calls intercepted by the interceptor into native kernel calls.

2. (ORIGINAL) An emulator as recited in claim 1, wherein the call-converter comprises a translator configured to translate a non-native paradigm for passing parameters into a native paradigm for passing parameters.

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324-9256  
F: 509.323-8979  
www.lee&hayes.com

**lee & hayes**

1           3.    (ORIGINAL)    An emulator as recited in claim 1, wherein the  
2 call-converter comprises a translator configured to translate non-native CPU  
3 instructions into native CPU instructions.

4  
5           4.    (ORIGINAL)    An emulator as recited in claim 1, wherein the  
6 call-converter comprises a translator configured to translate addresses from non-  
7 native length into native length.

8  
9           5.    (ORIGINAL)    An emulator as recited in claim 1, wherein the  
10 call-converter comprises an argument-converter configured to convert non-native  
11 argument format into native argument format.

12  
13           6.    (ORIGINAL)    An emulator as recited in claim 1, wherein the  
14 call-converter comprises a translator configured to translate words from non-  
15 native word size into native word size.

16  
17           7.    (ORIGINAL)    An emulator as recited in claim 1 further  
18 comprising a memory constrainer configured to limit addressable memory to a  
19 range addressable by non-native program modules.

20  
21           8.    (ORIGINAL)    An emulator as recited in claim 1 further  
22 comprising a shared-memory manager configured to manage memory space that is  
23 accessible to both native and non-native program modules.  
24  
25

lee & hayes  
421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324.9256  
F: 509.323.8978  
www.leeandhayes.com

1           9.    (ORIGINAL)    An emulator as recited in claim 1 further  
2 comprising a shared-memory manager configured to synchronize a native shared  
3 data structure with a non-native shared data structure.

4  
5           10. (PREVIOUSLY PRESENTED)    An emulator as recited in  
6 claim 1 further comprising a shared-memory manager configured to manage  
7 memory space that is accessible to both native and non-native program modules,  
8 wherein the shared-memory manager maps versions of process shared data  
9 structures (process SDSs) and versions of thread shared data structures (thread  
10 SDSs) between native and non-native program modules.

11  
12           11. (ORIGINAL)    An operating system on a computer-readable  
13 medium, comprising:

14           a native kernel configured to receive calls from native program modules;

15           a kernel emulator as recited in claim 1 configured to receive calls from non-  
16 native program modules.

17  
18           12. (ORIGINAL)    An operating system on a computer-readable  
19 medium, comprising:

20           a native kernel configured to receive calls from native APIs;

21           a kernel emulator as recited in claim 1 configured to receive calls from non-  
22 native APIs.

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324-9258  
F: 509.323-8979  
www.lee&hayes.com  
**lee & hayes**

1           **13. (ORIGINAL)**   A method of emulating a kernel for non-native  
2 program modules, the method comprising:

3           intercepting kernel calls from non-native program modules;

4           converting the intercepted non-native kernel calls into native kernel calls.  
5

6           **14. (ORIGINAL)**   A method as recited in claim 13, wherein the  
7 converting step comprises translating a non-native paradigm for passing  
8 parameters into a native paradigm for passing parameters.  
9

10          **15. (ORIGINAL)**   A method as recited in claim 13, wherein the  
11 converting step comprises translating non-native CPU instructions into native  
12 CPU instructions.  
13

14          **16. (ORIGINAL)**   A method as recited in claim 13, wherein the  
15 converting step comprises translating addresses from non-native length into native  
16 length.  
17

18          **17. (ORIGINAL)**   A method as recited in claim 13, wherein the  
19 converting step comprises translating words from non-native word size into native  
20 word size.  
21

22          **18. (ORIGINAL)**   A method as recited in claim 13 further  
23 comprising limiting addressable memory to a range addressable by non-native  
24 program modules.  
25

1           **19. (ORIGINAL)**   A method as recited in claim 13 further  
2 comprising synchronizing a native shared data structure with a non-native shared  
3 data structure.

4  
5           **20. (ORIGINAL)**   A method as recited in claim 13 further  
6 comprising mapping versions of process shared data structures (SDSs) between  
7 native and non-native program modules.

8  
9           **21. (ORIGINAL)**   A method as recited in claim 19, wherein a  
10 process SDS of a native program module includes a pointer to a process SDS of a  
11 non-native program module.

12  
13           **22. (ORIGINAL)**   A method as recited in claim 19, wherein a  
14 process SDS of a non-native program module includes a pointer to a process SDS  
15 of a native program module.

16  
17           **23. (ORIGINAL)**   A method as recited in claim 13 further  
18 comprising mapping versions of thread shared data structures (SDSs) data  
19 structure between native and non-native program modules.

20  
21           **24. (ORIGINAL)**   A method as recited in claim 22, wherein a  
22 thread SDS of a native program module includes a pointer to a thread SDS of a  
23 non-native program module.

24  
25

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324.9256  
F: 509.323.9879  
www.leeandhayes.com  
**lee & hayes**

Serial No.: 09/847,535  
Atty Docket No.: MS1-665us  
RESPONSE TO FINAL OFFICE ACTION DATED  
7/13/2005 UNDER 37 C.F.R. § 1.116

6


1014051524 O:\DOCS\MS1\0665US\005561.DOC  
att: Kasey C. Christie

1           **25. (ORIGINAL)**   A method as recited in claim 22, wherein a  
2 thread SDS of a non-native program module includes a pointer to a thread SDS of  
3 a native program module.

4  
5           **26. (ORIGINAL)**   A computer comprising one or more computer-  
6 readable media having computer-executable instructions that, when executed by  
7 the computer, perform the method as recited in claim 13.

8  
9           **27. (ORIGINAL)**   A computer-readable medium having computer-  
10 executable instructions that, when executed by a computer, performs the method  
11 as recited in claim 13.

12  
13           **28. (ORIGINAL)**   An operating system embodied on a computer-  
14 readable medium having computer-executable instructions that, when executed by  
15 a computer, performs the method as recited in claim 13.

16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
 421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324.9256  
F: 509.323.8979  
www.leeandhayes.com

1       **29. (PREVIOUSLY PRESENTED)**       A method comprising:  
2       determining whether an initiating program module is a native or non-native;  
3       if the initiating program is non-native:

4               limiting available memory to a range that is addressable by the non-  
5       native program module, that range of addressable memory being less than  
6       the available memory;

7               establishing non-native a version of a shared memory data structure  
8       that may be synchronized with a native version of the same shared memory  
9       data structure.

10  
11       **30. (ORIGINAL)**       A method as recited in claim 29 further  
12       comprising:

13               intercepting kernel calls from the non-native program module;  
14               converting the intercepted non-native kernel calls into native kernel calls.

15  
16       **31. (ORIGINAL)**       A method as recited in claim 29 further  
17       comprising emulating a non-native kernel for which kernel calls from the non-  
18       native program module are intended.

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324-9256  
F: 509.323-8979  
www.leeandhayes.com  
**lee & hayes**

1           32. (ORIGINAL) A computer comprising one or more computer-  
2 readable media having computer-executable instructions that, when executed by  
3 the computer, perform the method as recited in claim 29.

4  
5           33. (ORIGINAL) A computer-readable medium having computer-  
6 executable instructions that, when executed by a computer, performs the method  
7 as recited in claim 29.

8  
9           34. (ORIGINAL) A method comprising emulating a non-native  
10 kernel for a native computing platform so that kernel calls from non-native  
11 applications are translated into calls to a native kernel.

12  
13           35. (ORIGINAL) A method as recited in claim 34, wherein the  
14 emulating step comprises:

15 translating non-native CPU instructions into native CPU instructions;

16 translating addresses from non-native length into native length;

17 limiting addressable memory to a range addressable by non-native program  
18 modules.

19  
20           36. (ORIGINAL) A method as recited in claim 35, wherein the  
21 emulating step further comprises translating a non-native paradigm for passing  
22 parameters into a native paradigm for passing parameters.



1           **37. (ORIGINAL)**    A method as recited in claim 34, wherein the  
2 converting step further comprises translating words from non-native word size into  
3 native word size.

4  
5           **38. (ORIGINAL)**    A computer comprising one or more computer-  
6 readable media having computer-executable instructions that, when executed by  
7 the computer, perform the method as recited in claim 34.

8  
9           **39. (ORIGINAL)**    A computer-readable medium having computer-  
10 executable instructions that, when executed by a computer, performs the method  
11 as recited in claim 34.

12  
13           **40. (ORIGINAL)**    A kernel emulator configured to emulate a non-  
14 native kernel for a native computing platform so that kernel calls from non-native  
15 applications are translated into calls to a native kernel.

16  
17           **41. (ORIGINAL)**    An emulator as recited in claim 40, wherein the  
18 emulator comprises:

19           an instruction-translator configured to translate non-native CPU  
20 instructions into native CPU instructions;

21           an address-translator configured to translate addresses from non-native  
22 length into native length;

23           an memory constrainer configured to limit addressable memory to a range  
24 addressable by non-native program modules.

25

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324-9256  
F: 509.323-8979  
www.lee&hayes.com  
**lee & hayes**

Serial No.: 09/847,535  
Atty Docket No.: MS1-665us  
RESPONSE TO FINAL OFFICE ACTION DATED  
7/13/2005 UNDER 37 C.F.R. § 1.116

10

1014081524 0:100CSIMS110665US1805561.DOC  
atty: Kasey C. Chrissie

1           **42. (PREVIOUSLY PRESENTED)**       An operating system on a  
2 computer-readable medium, comprising:

3           a native kernel configured to receive calls from native program modules;  
4           a kernel emulator as recited in claim 40 configured to receive calls from  
5 non-native program modules.

6  
7           **43. (CANCELED)**

8  
9  
10          **44. (CANCELED)**

11  
12  
13          **45. (ORIGINAL)**   A kernel emulator for non-native program  
14 modules, the emulator comprising:

15          an interceptor configured to intercept kernel calls from non-native program  
16 modules;

17          a call-converter configured to convert non-native kernel calls intercepted by  
18 the interceptor into native kernel calls, wherein the call-converter comprises:

19               an instruction-translator configured to translate non-native CPU  
20 instructions into native CPU instructions;

21               an address-translator configured to translate addresses from non-  
22 native length into native length.

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324-9256  
F: 509.323-8979  
www.leeandhayes.com  
**lee & hayes**

1           **46. (ORIGINAL)**     An operating system on a computer-readable  
2 medium, comprising:  
3           a native kernel configured to receive calls from native program modules;  
4           a kernel emulator as recited in claim 45 configured to receive calls from  
5 non-native program modules.  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

421 West Riverside, Suite 500  
Spokane, WA 99201  
P: 509.324-9256  
F: 509.323-8979  
www.leeandhayes.com  
**lee & hayes**

Serial No.: 09/847,535  
Atty Docket No.: MS1-665us  
RESPONSE TO FINAL OFFICE ACTION DATED  
7/13/2005 UNDER 37 C.F.R. § 1.116

12

1014051524 0:\DOCS\MS1\0665US\055561.DOC  
by: Kasey C. Chrissie